

BEST AVAILABLE COPY

(43) Date of publication of application : 28.06.2002

GOLF 9/445

(71)Applicant :

(72)Inventor :

TSUNETOMI KUNIHICO

SUZUKI SHOJI

YAMADA TSUTOMU

YOKOYAMA TAKANORI

SAITO MASAHIKO

NAYA EKO

FUNAKI SATORU

ARITA YUTAKA

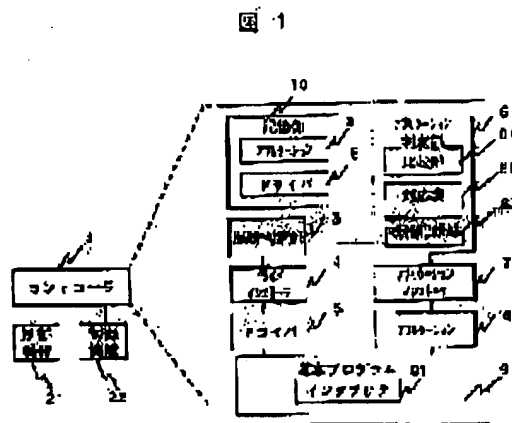
OKURA TAKANORI

NAKAMIGAWA TETSUAKI

(57)Abstract:

(57) Abstract:
PROBLEM TO BE SOLVED: To eliminate the need to install application corresponding to a connected controller whenever a new controller is connected.

SOLUTION: In the controller with which plural pieces of control equipment can be connected, an equipment judging part 3 to judge kinds of the connected control equipment, an application judging part 6 to receive an equipment judgment result of the control equipment judging part 3 and to judge application to be installed and an application installer 7 to install the application specified by the application judging part 6 are provided.



[Date of request for examination]

18.11.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

withdrawal

[Date of final disposal for application]

19.05.2006

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-182919

(P2002-182919A)

(43) 公開日 平成14年6月28日 (2002. 6. 28)

(51) IntCl.

G 0 6 F 9/445

識別記号

F I

G 0 6 F 9/06

テーマコード(参考)

6 1 0 L 5 B 0 7 6

6 5 0 B

審査請求 未請求 請求項の数 6 O L (全 10 頁)

(21) 出願番号

特願2000-379753(P2000-379753)

(22) 出願日

平成12年12月8日 (2000. 12. 8)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 恒富 邦彦

茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内

(72) 発明者 鈴木 昭二

茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内

(74) 代理人 100075096

弁理士 作田 康夫

最終頁に続く

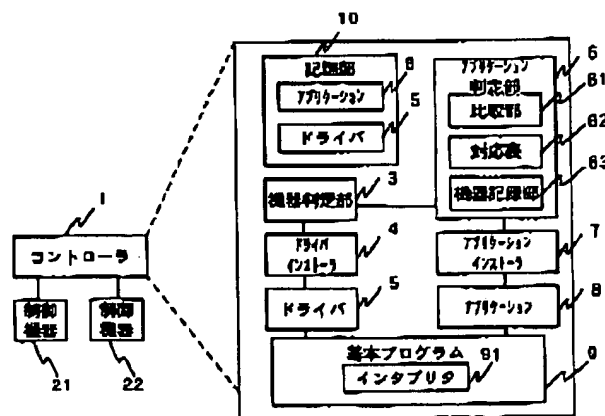
(54) 【発明の名称】 コントローラ及びアプリケーションインストール方法

(57) 【要約】

【課題】 コントローラには、多種多様の制御機器が接続されるが、これらを制御するアプリケーションの開発は、一般ユーザには困難である。また、制御機器の組み合わせの数だけアプリケーションをコントローラと抱き合わせて提供しても、ユーザが正しく選択し、インストールするのは困難である。

【解決手段】 複数の制御機器が接続可能なコントローラにおいて、接続された制御機器の種別を判定する機器判定部3と、該制御機器判定部3の機器判定結果を受信し、インストールすべきアプリケーションを判定するアプリケーション判定部6と、該アプリケーション判定部6に指定されたアプリケーションをインストールする、アプリケーションインストーラ7を備える。

図 1



(2)

特開2002-182919

2

【特許請求の範囲】

【請求項1】複数の制御機器が接続可能なコントローラにおいて、

接続された前記制御機器の種類または個数、もしくは種類及び個数を判定する機器判定手段と、

前記機器判定手段にて判定された種類または個数、もしくは種類及び個数に基づいて、前記種類または個数、もしくは種類及び個数に対応するアプリケーションを判定するアプリケーション判定手段と、

前記アプリケーション判定手段で判定されたアプリケーションをインストールするインストール手段を有するコントローラ。

【請求項2】請求項1記載のコントローラにおいて、前記機器判定手段は、前記制御機器が接続された時、自動的に前記制御機器の種別を判定し、前記インストール手段は、判定されたアプリケーションを自動的にインストールするコントローラ。

【請求項3】請求項1記載のコントローラにおいて、前記アプリケーション判定手段は、前記制御機器の種類または個数、もしくは種類及び個数と、記憶されたアプリケーションとが対応付けた対応データが記憶された対応記録手段と、前記対応記録部に記憶された対応データに基づいて、接続された制御機器に対応するアプリケーションを選択する選択手段とを有するコントローラ。

【請求項4】複数の制御機器が接続可能なコントローラにおいて、

ネットワークと接続するためのインタフェース手段と、接続された前記制御機器の種別を判定する機器判定手段と、

前記機器判定手段にて判定された種別に基づいて、前記種別に対応するアプリケーションを判定するアプリケーション判定手段と、

前記アプリケーション判定手段で判定されたアプリケーションを前記ネットワークを介して入力し、入力された前記アプリケーションをインストールするインストール手段を有するコントローラ。

【請求項5】請求項4記載のコントローラにおいて、少なくとも前記制御機器に対応するアプリケーションが記憶された記憶部とネットワークを介して接続されるコントローラ。

【請求項6】複数の制御機器を接続可能なコントローラに制御機器の接続の可否及び前記制御機器の種類または個数、もしくは種類及び個数を検知し、制御機器が接続された場合、検知された制御機器の種類または個数、もしくは種類及び個数に基づいてアプリケーションを判定し、

自動的に前記判定されたアプリケーションをインストールするアプリケーションインストール方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、機器の監視や制御に用いられるコントローラへのアプリケーションのインストール方法及びそれを用いたコントローラ、計算機に関する。

【0002】

【従来の技術】従来のコントローラの構成を、図2に示す。コントローラのソフトウェアは、基本プログラム9、ドライバ5、アプリケーション8から構成されている。

【0003】基本プログラム9は、一般にモニタ、またはOS (Operating System)と呼ばれる部分である。基本プログラム9は、コントローラ1のハードウェアの管理や、プログラム（後述するドライバ5、アプリケーション8）を実行する機能を提供する。また、コントローラがネットワークに接続されている場合、ネットワークのプロトコルを提供する。OSの機能については、「UNIX 4.3 BSD の設計と実装」（中村明 他：丸善）に詳しく解説されている。

【0004】ドライバ5は、制御機器21、22の制御を行うためのプログラムである。ドライバ5は、制御機器21、22の初期化や、制御機器21、22へのリード、ライトを行う基本的な関数を、基本プログラム9やアプリケーション8に提供する。制御機器21、22は、これを動作させるためのレジスタを持っており、ドライバ5はこのレジスタをライトすることで制御機器21、22を動作させる。また、ドライバ5は、レジスタをリードすることにより、制御機器21、22の状態を確認する。一般のコントローラでは、ドライバは基本プログラムに最初から含まれている。したがって、コントローラのユーザは、ドライバのインストールを行う必要は無いが、新たな、制御機器をコントローラに追加する場合、基本プログラム自体の入れ替えが必要になる。これを解決する方法として、PC（パーソナルコンピュータ）の分野では、PnP（プラグアンドプレイ）、UPnP（ユニバーサルプラグアンドプレイ）技術が導入されている。すなわち、機器判定部33を持ち、判定結果により必要なドライバ5をドライバインストール4により自動的にインストールする方法である。この技術は、コントローラにも適用可能である。これを適用したコントローラを、図3に示す。

【0005】ここで、UPnPの概要について説明しておく。UPnPは、接続機器と計算機が通信し合うプロトコルを規定している。このプロトコルを、接続機器と計算機の双方が内蔵する。計算機はこのプロトコルを用いて、接続機器の種類（UPnPではデバイスタイプ）や、接続機器が提供する機能（UPnPではデバイスサービス）を問い合わせることができる。具体的には、コントローラが、SSDP (Simple Service Discovery Protocol) のM-SEARCHメソッドを制御機器にマルチキャストする。M-SEARCHを受信した制御機器は、U

(3)

特開2002-182919

3

UIDとよばれるデバイス番号（本発明の制御機器名称）を応答する。UPnPの詳細仕様については、「Universal Plug and Play Device Architecture Version 1.0(ユニバーサル・プラグ・アンド・プレイ・デバイス・アーキテクチャ)」を参照されたい。

【0006】アプリケーション8は、ユーザが制御機器21, 22を組み合わせて行いたい、制御を行うプログラムである。たとえば、ユーザはカードリーダー204により開閉する自動扉システムを構築するとする。このとき、ユーザは図16のコントローラ102のように、コントローラ1にカードリーダー204と自動扉205が接続される。ドライバ5は、カードリーダー204、および自動扉205を動作させる基本的な関数を提供する。アプリケーション8は、ドライバ5の関数を用いて、ユーザの行いたい制御を実現する。すなわち、アプリケーション8がカードリーダー204の判定結果をカードリーダードライバによりリードする。もし、正しいカードであった場合は、アプリケーションが自動扉ドライバに自動扉を開けるように命令する。このように、アプリケーションは、各種ドライバを結合すると同時に、「カードが有効ならば自動扉を開ける」といった論理制御を行う。コントローラのアプリケーションは、「ラダー」「ファンクションブロック」「IL（インストラクションランゲージ：Instruction Language）」などのコントローラ専用言語を用いて、ユーザにより作成されている。この言語は、インタプリタにより解釈・実行される。また、アプリケーションが「C言語」により記述され、コンパイラによりオブジェクトコードに変換されたオブジェクトファイルであることがある。この場合はプロセッサにより直接実行される。「ラダー」「ファンクションブロック」「IL」などのコントローラ専用言語は「新しいプログラマブルコントローラのプログラミング」（関口隆著：コロナ社）や「アイ・トリプル・イー61131-3（IEEE61131-3）」にて定義されているので参照されたい。

【0007】

【発明が解決しようとする課題】最近、コントローラが小型化・低価格化したことから、FA（ファクトリーオートメーション）やビル管理だけでなく、一般家庭でも使用される可能性が出てきた。前述の通り、コントローラを使用するためには、アプリケーションをユーザがプログラミングする必要があるが、一般ユーザは、「ラダー」「ファンクションブロック」といった、プログラミング言語に関する知識が少ない。したがって、一般ユーザが、コントローラのプログラミングを行うのは困難である。そこで、コントローラのプログラミング言語を使用したアプリケーションを販売して、ユーザにインストールさせることが考えられる。しかし、この方式にも問題がある。ユーザは、用途に応じて、制御機器を多種、多様なコントローラに接続する。それらの組み合わせに

4

応じてアプリケーションをユーザが選択して購入するのは困難な作業となる。

【0008】

【課題を解決するための手段】本発明は、複数の制御機器が接続可能なコントローラにおいて、接続された制御機器の種類または個数、もしくは種類及び個数を判定する機器判定手段と、判定された種類または個数、もしくは種類及び個数に基づいて、種類または個数、もしくは種類及び個数に対応するアプリケーションを判定するアプリケーション判定手段と、判定されたアプリケーションをインストールするインストール手段を有する。

【0009】また、機器判定手段は、制御機器が接続された時、自動的に制御機器の種別を判定し、インストール手段は、判定されたアプリケーションを自動的にインストールする構成とする。

【0010】また、アプリケーション判定手段は、制御機器の種類または個数、もしくは種類及び個数と、記憶されたアプリケーションとが対応付けた対応データが記憶された対応記録手段と、対応記録部に記憶された対応データに基づいて、接続された制御機器に対応するアプリケーションを選択する選択手段とを有する。

【0011】また、複数の制御機器が接続可能なコントローラにおいて、ネットワークと接続するためのインタフェース手段と、接続された制御機器の種別を判定する機器判定手段と、機器判定手段にて判定された種別に基づいて、種別に対応するアプリケーションを判定するアプリケーション判定手段と、判定されたアプリケーションをネットワークを介して入力し、入力されたアプリケーションをインストールするインストール手段を有する。

【0012】また、少なくとも前記制御機器に対応するアプリケーションが記憶された記憶部とネットワークを介して接続される構成とする。

【0013】また、複数の制御機器を接続可能なコントローラに制御機器の接続の可否及び制御機器の種類または個数、もしくは種類及び個数を検知し、制御機器が接続された場合、検知された制御機器の種類または個数、もしくは種類及び個数に基づいてアプリケーションを判定し、自動的に判定されたアプリケーションをインストールする方法とする。

【0014】

【発明の実施の形態】以下、図1～図16を使用して、実施例を示す。

（実施例1）実施例1では、コントローラ単体で本発明を実施した例を示す。図1に、本実施例の構成を示す。図9にコントローラハードウェアの構成を示す。

【0015】図9を用いて、コントローラハードウェアを説明する。コントローラ1は、プロセッサ104、メモリ105、I/Oコントローラ107、LANコントローラ106から成る。メモリ105は、コントローラ

(4)

特開2002-182919

5

ソフトウェアを保存する。プロセッサ104は、コントローラソフトウェアの機器判定部33、ドライバインストーラ4、基本プログラム9、アプリケーション、アプリケーションインストーラ7、比較部61の各プログラムをリードして実行する。LANコントローラ106は、広域ネットワークに接続するために使用される。LANコントローラ106が、ネットワークからの受信データをプロセッサ104に送る。ネットワークへ送付するデータをプロセッサ104より受け付け、ネットワーク30上に転送する。ネットワークを使用しない場合は、LANコントローラ106を持つ必要はない。制御機器21は、コントローラから制御される機器である。制御機器21は、一般にレジスタと呼ばれるバッファを持ち、コントローラ1からライト・リード可能である。レジスタには、ステータスレジスタとコントロールレジスタの2種類がある。制御機器21は、コントロールレジスタにライトされたデータ内容にしたがって動作する。そして、動作状態、または動作した結果を、ステータスレジスタに表示する。制御機器としては、工業分野では、センサ、インバータ、サーボアンプ、モータ、コントローラ、スイッチ、タッチパネル、液晶パネルコントローラ、アクチュエータなど、FAラインに使用される機器がある。また、家庭用分野では、ビデオカメラ、ビデオ、テレビ、電話、インタフォン、冷蔵庫、洗濯機、掃除機、電子レンジ、ドアの鍵、ウォシュレット、照明機器、エアコン、電機湯沸し機、ガス湯沸し機、炊飯器、時計など家電品が考えられる。また、オフィスはビル管理分野では、カードリーダー204、ファックス、コピー、電話、スキャナ、パソコン、電流計、電力計、ブレーカなどのOA機器やビル管理器具が考えられる。I/Oコントローラ107は、プロセッサ104からのリード・ライト命令を、制御機器21のレジスタに反映する。リード命令の場合、制御機器21のレジスタの内容をプロセッサ104に送信する。ライト命令の場合、I/Oコントローラ107は、制御機器21のレジスタにデータをライトする。

【0016】次に、図1を用いて、コントローラソフトウェアの構成を説明する。

【0017】機器判定部33は、コントローラに接続された制御機器21の種類を判定するソフトウェアである。機器判定部33は、UPnPのような既存の技術を用いて、個々の制御機器についての判定を行うことができる。機器判定部33は、接続が確認された制御機器の種別を、アプリケーション判定手段に通知する。具体的には、機器記録部63に記録する。また、制御機器の個数を記録する。ドライバインストーラ4が存在する場合は、そちらにも制御機器の種別を通知する。

【0018】ドライバは、制御機器の初期化や、制御機器をアプリケーション、基本ソフトウェアから使用されるため、統一インタフェースを用意する。たとえば、前

6

出の「UNIX4.3BSDの設計と実装」に記述されているように、read(リード)、write(ライト)、ioctl(制御)など、あらかじめ決められた関数(以下統一インタフェースとよぶ)を用意しておき、これを使用するアプリケーションからは、その関数を使ってコールするようプログラミングする。これにより、ドライバが、ドライバインストーラ4により、動的にインストールされたととしても、関数名が異なるためにアプリケーションとドライバがリンクできないということは発生しない。

【0019】ドライバインストーラ4は、ドライバ5を動的にインストールするソフトウェアである。ドライバ5のインストールは、以下の3つの手順に分けられる。まず、ドライバプログラムを記録できるだけのメモリを割り当てる。これは基本プログラム9のメモリ管理を使用することで実現できる。また、インストーラ自身がメモリ管理を持っていてもよい。次に、記憶部10から必要なドライバをリードし、割り当てられたメモリにライトする。最後に、統一インタフェースの関数アドレスを、基本プログラム9に登録する。なお、ドライバを基本プログラム9に静的にリンクする場合は、該ドライバインストーラ4は必要ない。本実施例でも特に必須の部分ではない。

【0020】アプリケーション判定部6は、コントローラにインストールするアプリケーションを選択するプログラムである。アプリケーション判定部6は、機器記録部63、対応表62、比較部61から構成される。

【0021】機器記録部63の構成を図6に示す。機器記録部63は、接続機器の種類631と個数632を保存する。図6の種類631は、接続機器の名称が記録されているが、あらかじめ機器ごとに割り与えられた一意な番号でもよい。機器記録部63は、機器判定部33によりライトされる。

【0022】対応表62の構成を図7に示す。対応表62は、アプリケーション名と、そのアプリケーションが使用する、制御機器の個数の対応を記録するN×Mの配列である。Nは制御機器の種別個数、Mはアプリケーションの個数である。各列に、制御機器の個数622、623、624、625、626と、その機器が接続されているときにインストールされるアプリケーションの名称621が記録される。ただし、1列目は、制御機器名称が記録されている。列番号と制御機器が1対1に対応していれば、制御機器名称は特に必要ない。図7には、「ビデオカメラ」「ブザー」など制御機器名が直接記録されているが、制御機器にあらかじめ割り当てられたID(番号)でもよい。アプリケーション名も同様に、アプリケーションにあらかじめ割り当てられたID(番号)でもよい。2～N行目には、各制御機器の個数を示す数字が記録されるが、数字以外に「1以上」「2以下」「0以上2以下」といった、数値範囲を記録できる。

(5)

特開 2002-182919

7

【0023】比較部61は、該コントローラにインストールするアプリケーションを選択するプログラムである。比較部61は、機器記録部63より現在の制御機器の種類と個数をリードし、対応表62から制御機器の条件が合致するアプリケーションを検索する。

【0024】機器記録部63、対応表62、比較部61の関係を、図4に示す。機器記憶部10は、機器判定部33によってライトされ、選択部によってリードされる。また、対応表62は、あらかじめ初期化されており、比較部61によりリードされる。これらの動作は、

後述する図11のフローチャートにおいて説明する。

【0025】以上が、アプリケーション判定部6の説明である。

【0026】アプリケーションインストーラ7は、アプリケーションを動的にインストールするプログラムである。アプリケーションのインストール手順は、以下の2つの手順に分けられる。最初に、アプリケーションプログラムを記録できるだけのメモリを割り当てる。これは基本プログラム9のメモリ管理を使用することで実現できる。また、インストーラ自身がメモリ管理を持ってもよい。次に、記憶部10から必要なアプリケーションをリードし、割り当てられたメモリにライトする。

【0027】アプリケーション8は、ユーザが制御機器を組み合わせて行いたい、論理制御を行うプログラムである。コントローラのアプリケーションは、プロセッサ104が直接実行できるオブジェクトコードで記述されている場合と、ラダー言語のようにインタプリタ91言語で記述される場合がある。インタプリタ言語の場合は、基本プログラム9に含まれるインタプリタ91がアプリケーションを解釈し、オブジェクトコードに変換してプロセッサ104が実行する。後の動作説明ではアプリケーションがラダー言語で記述されている場合を扱うが、アプリケーションの実行のなされかた以外に基本的な違いはない。アプリケーションから、制御機器を動作させる場合は、ドライバの関数を呼び出して制御する。ドライバの呼び出しは、基本プログラム9を介して、ドライバ関数のアドレスを得ることよりコールできる。

【0028】以上が、図1の実施例の説明である。

【0029】次に、図11～図14のフローチャートを用いて、アプリケーションの自動インストール動作を説明する。

【0030】図11に機器判定部33の動作のフローチャートを示す。本実施例では、機器判定部33は、コントローラの電源投入時に自動的に開始する。しかし、コントローラにリセットボタンがついている実施例の場合は、リセットごとに機器判定部33を実行することができる。

【0031】機器判定部33は、実行を開始すると、まず、接続されている制御機器を検索する(ステップ1001)。制御機器の検索は、従来技術で実現できる。た

8

例えばUPnPでは、SSDP(Simple Service Discovery Protocol)のM-SEARCHメソッドをコントローラが制御機器にマルチキャストする。M-SEARCHを受信した制御機器は、UUIDとよばれるデバイス番号(本発明の制御機器名称)を応答する。これにより、接続されている制御機器を調査できる。

【0032】次に、制御機器が判定できたか調べる(ステップ1002)。もし制御機器が発見できれば、同じ名称の制御機器が、制御機器記録部63に記録されているか調べる(ステップ1003)。もし、記録されていれば、新たに制御機器の名称を記録する。その後、制御機器記録部63の個数を1増加する(ステップ1004)。その後、他の制御機器を探すために、ステップ1001に戻る。

【0033】ステップ1002において、制御機器が新たに発見できなければ、アプリケーション判定部6を起動する(ステップ1005)。その後、機器判定部33を停止する(ステップ1006)。アプリケーション判定部6を起動する方法の1つは、基本プログラム9の機能を使用して、アプリケーション判定部6をタスク(またはプロセス)として生成し、実行を開始させることである。

【0034】以上の機器判定部33の処理により、機器記録部63の記録が完了する。例えば図5のコントローラ102の場合、図6の内容の機器記録部63となる。

【0035】以上で、図11の説明を終わる。

【0036】図12にアプリケーションインストーラ7のフローチャートを示す。アプリケーションインストーラ7は、機器判定部33から起動される。アプリケーションインストーラ7は、まず、対応表62の2番目のエントリ(図6の実施例では、省エネアプリケーションのエントリ)を選択する(ステップ2001)。次に、選択したエントリの制御機器の種類、個数と、機器記録部63の種類、個数が合致するか比較する(ステップ2002)。もし、合致しなければ、対応表62の次のエントリを選択し(ステップ2003)、ステップ2002に戻る。ステップ2002において、制御機器の種類、個数が合致すれば、現在選択しているアプリケーションをインストールすべきアプリケーションとして選択し、アプリケーションインストーラ7にインストールを依頼する(ステップ2004)。その後、アプリケーション判定部6の処理を終了する(ステップ2005)。アプリケーションインストーラ7への依頼の実現手段としては、基本プログラム9の機能を使用して、アプリケーションインストーラ7をタスク(またはプロセス)として生成し、実行を開始させることである。

【0037】以上の機器判定部33の処理により、アプリケーションの選択が完了する。例えば図5のコントローラ102の場合、対応表62が図7の場合は、制御機器の個数が「認証付き自動扉アプリケーション82」と

50

(6)

特開2002-182919

9

合致するので、「認証付き自動扉アプリケーション82」が選択される。

【0038】以上で、図12の説明を終わる。

【0039】図13にアプリケーションインストーラ7のフローチャートを示す。アプリケーションインストーラ7は、アプリケーション判定部6より起動される。初めに、アプリケーションインストーラ7は、アプリケーションを記録部からリードする(ステップ3001)。次に、アプリケーションのサイズを測定し、同等のサイズのメモリを割り当てる(ステップ3002)。ステップ3001のアプリケーションリード、およびステップ3002のサイズ測定は、基本プログラム9のファイルシステムにより実現する。また、ステップ3002のメモリ割り当ては、基本プログラム9のメモリ管理システムコールにより実現する。ステップ3002後、アプリケーションを、割り当てられたメモリにライトし、処理を終了する(ステップ3003)。

【0040】以上で、アプリケーションの自動インストール動作説明を終わる。

【0041】次に、上記アプリケーションインストーラ7でインストールしたアプリケーションの動作について説明する。前記図1の説明にて示した、認証付き自動扉アプリケーション82を例に説明する。

【0042】図16に認証付き自動扉アプリケーション82が制御する、制御機器の構成を示す。コントローラにカードリーダ204と、自動扉制御機器が接続されている。自動扉制御機器には、自動扉モータ2051と自動扉スイッチ2052がつけられている。自動扉スイッチ2052は、扉前のマットに仕掛けられており、マット上に通行人が立つと、オンになる。すると、自動扉モータ2051が、一定時間自動扉2053を開ける。

【0043】図14に認証付き自動扉アプリケーション82のラダープログラムを示す。本プログラムは、自動扉スイッチ2052がオンであり、かつ、カードリーダ204が正当なカードと認識したとき、自動扉を開けるように制御する。プログラム中、CR1はカードリーダ204の状態を示す。カードリーダ204がリードしたカードが、正当なカードならば、1、不正カードであれば0になる。SW1は自動扉スイッチ2052の状態を示す。オン状態のとき1、オフ状態のとき0となる。DR1は、ドアの開放命令を示す。1のとき、自動扉モータ2051が自動扉2053を開放する。

【0044】ラダー文法により、認証付き自動扉アプリケーションプログラムを解釈すると、「CR1とSW1の論理積をとり、DR1にライトする」という意味になる。これを、インタプリタ91が解釈し実行するのであるが、インタプリタ91動作の説明上、これをILに変換したプログラム(図15)により説明する。

【0045】以下、認証付き自動扉アプリケーション82を実行する時の、インタプリタ91の動作を説明す

10

る。インタプリタ91は、第一の命令(LD CR1)をリードする。LDはアキュムレータにCR1をロードする命令である。そこで、インタプリタ91は、自動扉ドライバのread関数を用いてCR1をリードし、アキュムレータにライトする。次に、インタプリタ91は、第二の命令(AND SW1)をリードする。ANDは、アキュムレータとSW1の論理積をとり、アキュムレータに格納する命令である。そこで、インタプリタ91は、カードリーダ204ドライバのread関数を用いて、SW1をリードし、アキュムレータと論理積をとり、アキュムレータにライトする。最後に、インタプリタ91は第三の命令(ST DR1)をリードする。STは、アキュムレータをDR1にストアする命令である。そこで、インタプリタ91は、自動扉ドライバのwrite関数を用いて、アキュムレータの内容をDR1にライトする。以上のインタプリタ91の動作により、CR1とSW1が1の時のみDR1を1にすることがわかる。

【0046】以上で、アプリケーションインストーラ7でインストールしたアプリケーションの動作説明を終わる。

(実施例2) 実施例1では、アプリケーションが認証付き自動扉アプリケーション82を例に説明したが、これは、図7に示すとおり、他のアプリケーションが使用していない制御機器のみを示している。したがって、アプリケーションを選択するときに、アプリケーション判定部6は、機器記録部63と対応表62の機器名称のみを比較すればよい。具体的には、ステップ1004削除、ステップ2002を「機器名称が合致するか?」に変更することができる。

【0047】逆に、機器名称と種別の両方の調査が必要な場合は、複数のアプリケーションが、同じ機器名称のものを使用するときである。たとえば、図5のコントローラ101のようにコントローラにビデオカメラとTVインタフォンが接続され、ビデオカメラの映像をテレビインタフォンに写す場合、ビデオカメラの台数により、インストールするアプリケーションが異なる。ビデオカメラが1台のときは、テレビインタフォンに1つの画面のみを表示する「シングルスクリーン表示アプリケーション81」をインストールする。ビデオカメラが2台以上のときは、テレビインタフォンに2画面以上を表示する「マルチスクリーン表示アプリケーション83」をインストールする。

(実施例3) コントローラに接続される制御機器(あるいは、アプリケーション判定部6がサポートする制御機器)の種類が増えると、対応表62のNが増加する。したがって、Nが増えるたびに、対応表62を更新する必要がある。また、サポートする制御機器が増加すると、これに対応して、サポートするアプリケーションが増加する。対応表62の更新や、アプリケーションの追加を、コントローラごとに行うのは、コストが高い。

(7)

特開 2002-182919

11

【0048】これを解決する実施例として、図8に示すように、アプリケーションと対応表62をネットワーク上の計算機に置き、複数のコントローラからこれらのデータを共有する方式がある。データの共有は、ネットワークファイルシステムを用いて実現できる。ネットワークファイルシステムは、ネットワーク上の計算機のファイルを自計算機上のファイルと同様のシステムコールを用いてリード、ライトできるようにする技術である。ネットワークファイルシステムを用いれば、計算機の対応表62・アプリケーションを、コントローラ上の機器判定部33、アプリケーション判定部6、アプリケーションインストーラ7から、ネットワークを意識することなくリードライトできる。サポートする、制御機器、アプリケーションを増加するときは、計算機1箇所の対応表62、アプリケーションを変更するのみでよい。アプリケーションインストーラ7、機器判定部33、アプリケーション判定部6を変更する必要はない。

【0049】また、他の解決方法として、図10に示すように、アプリケーション判定部6全てを他の計算機上に持ち、コントローラからこの判定部を使用する方式がある。この場合、機器判定部33からアプリケーション判定部6の起動と、アプリケーション判定部6からアプリケーションインストーラ7の起動は、実施例1の方式では実現できない。そこで、これらの起動処理を、リモートプロシージャコールに置きかえる。リモートプロシージャコールは、ネットワーク上の他の計算機のシステムコールを、自計算機からコールする技術である。リモートプロシージャコールにより、アプリケーション判定部6、機器判定部33、アプリケーションインストーラ7が、おのおの別の計算機にあることを意識せずにコールできる。

【0050】

【発明の効果】ユーザが、接続された制御装置に対応するアプリケーションを新しい制御装置を接続する度にインストールする必要がない。

【図面の簡単な説明】

【図1】本発明の実施例1の構成を示す。

【図2】従来例の構成を示す。

【図3】ドライバのプラグアンドプレイをサポートした従来例の構成を示す。

【図4】本発明の機器判定部3、アプリケーション判定部6、アプリケーションインストーラ7の関係を示す。

【図5】ネットワークの計算機上にアプリケーションと対応表62をもつ場合の、本発明の実施例の構成を示す。

【図6】機器記録部63の構成を示す。

【図7】対応表62の構成を示す。

【図8】ネットワークの計算機上にアプリケーションと

12

対応表62をもつ場合の、本発明の実施例の構成を示す。

【図9】コントローラハードウェアの構成を示す。

【図10】ネットワークの計算機上にアプリケーション判定部6とアプリケーションをもつ場合の、本発明の実施例の構成を示す。

【図11】機器判定部3のフローチャートを示す。

【図12】アプリケーション判定部6のフローチャートを示す。

【図13】アプリケーションインストーラ7のフローチャートを示す。

【図14】認証付き自動扉アプリケーション82のラダープログラムを示す。

【図15】認証付き自動扉アプリケーション82のILプログラムを示す。

【図16】認証付き自動扉のハードウェア接続構成を示す。

【符号の説明】

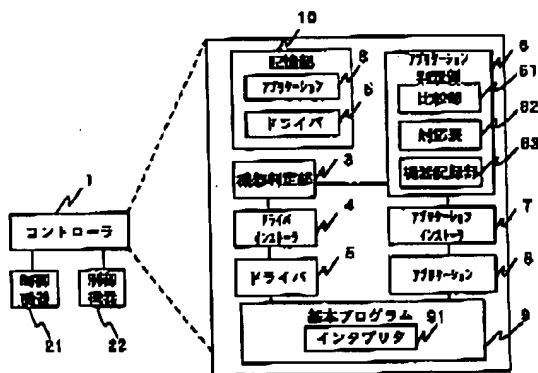
1, 101, 102, 103…コントローラ、3…機器判定部、4…ドライバインストーラ、5…ドライバ、6…アプリケーション判定部、7…アプリケーションインストーラ、8…アプリケーション、9…基本プログラム、10…記憶部、11…計算機、21, 22…制御機器、30…ネットワーク、40…認証付き自動扉アプリケーション82のラダープログラム、41…認証付き自動扉アプリケーション82のILプログラム、61…比較部、62…対応表、63…機器記録部、81…シングルスクリーン表示アプリケーション、82…認証付き自動扉アプリケーション、83…マルチスクリーン表示アプリケーション、91…インタプリタ、104…プロセッサ、105…メモリ、106…LANコントローラ、107…I/Oコントローラ、201, 202…エアコン、203…温度センサ、204…カードリーダ、205…自動扉、206, 207…ビデオカメラ、208…テレビインタフォン、621…対応表62のアプリケーション名称、622, 623, 624, 625, 626…対応表62の制御機器の個数、631…機器記録部63の種別、632…機器記録部63の個数、621…対応表62のアプリケーション名称、622, 623, 624, 625, 626…対応表62の制御機器の個数、1001, 1002, 1003, 1004, 1005, 1006…機器判定部3のフローのステップ、2001, 2002, 2003, 2004, 2005…アプリケーション判定部6のフローのステップ、2051…自動扉モータ、2052…自動扉スイッチ、2053…自動扉、3001, 3002, 3003…アプリケーションインストーラ7のフローのステップ。

(8)

特開2002-182919

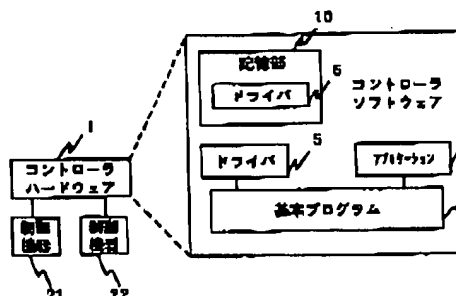
【図1】

図 1



【図2】

図 2



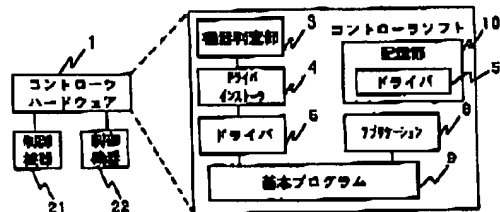
【図6】

図 6

番号	名称
631	電源部
632	制御部

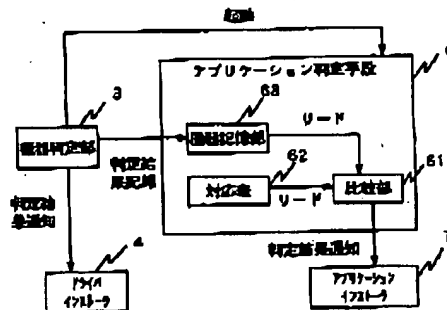
【図3】

図 3



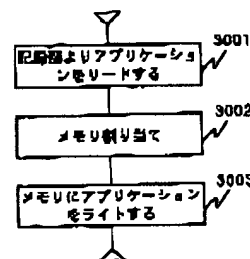
【図4】

図 4



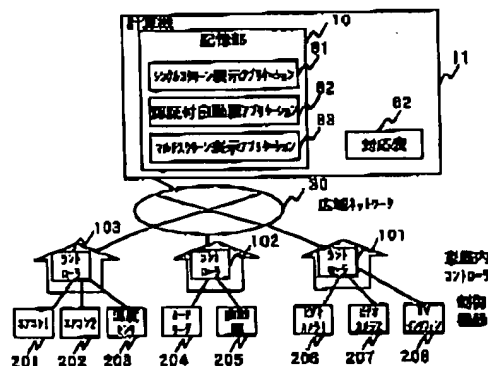
【図13】

図 13



【図5】

図 5



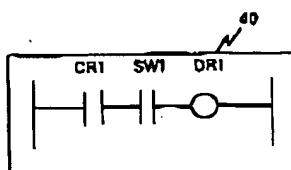
【図7】

図 7

アプリケーション名	821	822	823	824	825	826
音楽アプリケーション	1以上	1以上	0	0	0	0
画像付自動運転アプリケーション	0	0	1	1	0	0
画像付表示アプリケーション	2以上	0	0	0	0	1
画像付表示アプリケーション	1	0	0	0	0	1

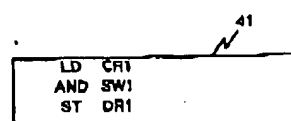
【図14】

図 14



【図15】

図 15

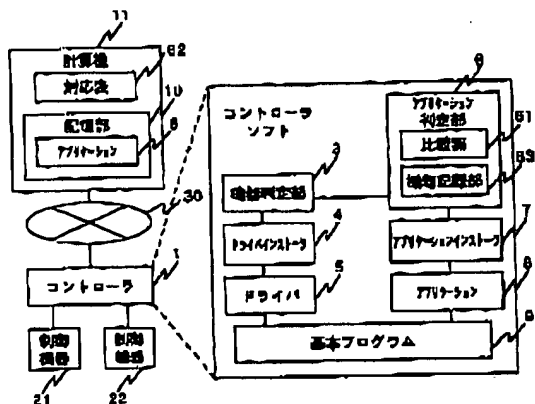


(9)

特開2002-182919

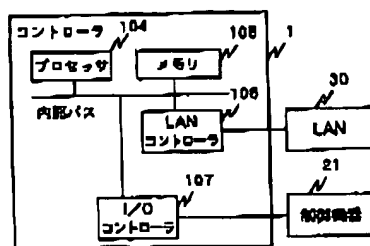
【図8】

図 8



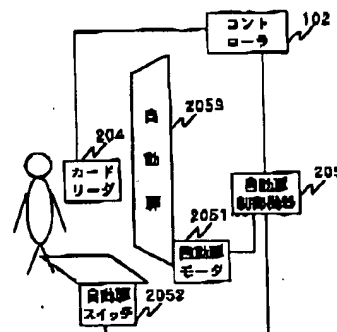
【図9】

図 9



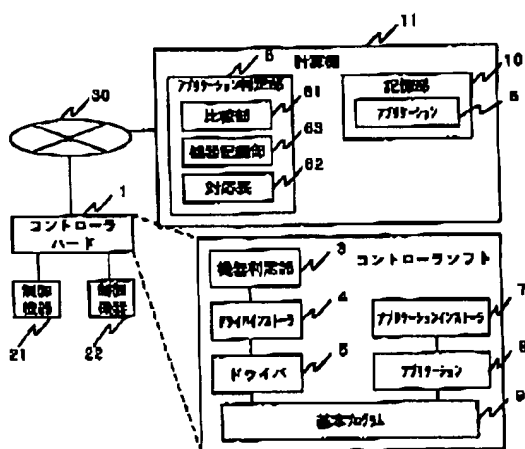
【図16】

図 16



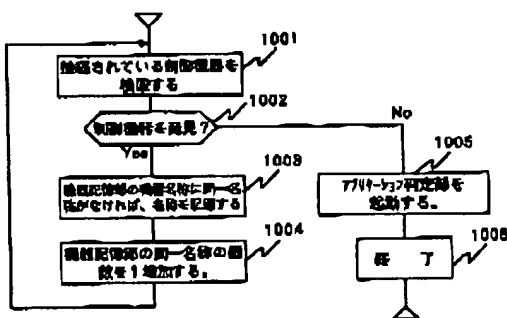
【図10】

図 10



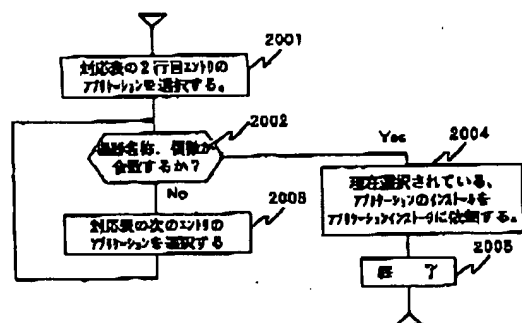
【図11】

図 11



【図12】

図 12



(10)

特開2002-182919

フロントページの続き

(72)発明者 山田 勉
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 横山 孝典
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 齊藤 雅彦
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 納谷 英光
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 船木 覚
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 有田 裕
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 大倉 敬規
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

(72)発明者 中三川 哲明
茨城県日立市大みか町七丁目1番1号 株
式会社日立製作所日立研究所内

Fターム(参考) 5B076 AA01 AB17